Thought Crimes and Profanities whilst Programming

Juho Leinonen
University of Helsinki
Helsinki, Finland
juho.leinonen@helsinki.fi

Arto Hellas
University of Helsinki
Helsinki, Finland
arto.hellas@cs.helsinki.fi

ABSTRACT

Where should we draw the line of inappropriate conduct on a course that is given freely to anyone? If an individual starts profusely swearing on a lecture, they are most likely expelled from the class or even from the course. But what if they do it outside the lecture amongst their classmates, amongst a group of anonymous individuals – or by themselves? In this article, we study how students use profanities in source code when they are completing programming assignments on a massive open online course (MOOC). We examine how common it is to curse in source code as well as whether specific assignments incite more cursing than others. Additionally, we investigate differences between participants with regards to cursing. Our results indicate that a considerable amount of participants write curse words whilst programming, but most clean their code for the final submission. The data also shows that there are different degrees of profanity in use, ranging from quite inoffensive words to offensive racial slurs. Finally, we discuss options that one may take when individuals who swear are identified, starting from rescinding their right to study.

CCS CONCEPTS

• **Social and professional topics** → *Codes of ethics*; *Censorship*; *Computer science education*;

KEYWORDS

profanity, thought crimes, student performance, student behavior, source code snapshots, educational data mining

1 INTRODUCTION

The dystopian novel Nineteen Eighty-Four by George Orwell [18] focuses on a society, where even the thoughts of the citizens are controlled. Specifically assigned thought police – *Thinkpol* – record and monitor interactions of the populace through various means, looking for opinions that challenge the views of the ruling party. This control and monitoring extends to the subconscious, as even stress and words that are uttered whilst asleep are monitored.

Although the novel is a work of fiction, the possibility for such surveillance is a reality [9, 23]. For example, Facebook records the comments that individuals write but do not post [5], Google records whatever is spoken to it [10], and commercial organizations profile users to determine their interests. This trend is not only visible in organizations, but also in the lives of individuals – for example, many wear devices that gather information on physical activities and provide suggestions based on the data.

Such surveillance data can change the life of a person drastically. Whilst sports tracking devices typically focus on self-improvement, recent news has witnessed the use of traces of human behavior in other ways as well. As an example, a group of newly admitted Harvard students had their admissions rescinded after posting offensive memes in a closed Facebook group [17].

In this article, we take the role of an imaginary thought police, and use data from a freely available online course to identify individuals who act against the norms of the society. More specifically, we study participants in a massive open online course on programming, and look for socially offensive language in their programming process. We then identify course-specific factors such as assignment handouts that may influence the use of profanities, and subsequently, study the differences between participants who curse and those who do not.

Our analysis covers both data intended as final products – submissions – as well as the working process – snapshots. The use of such data is popular within the computing education research domain where it is used, among other things, for predicting course outcomes [1, 14]. The studies that are closest to the study presented here are studies that look into affect – for example, Rodrigo et al. have used programming process data to identify affective states such as frustration [19], and Ihantola et al. have used such data to automatically assess the difficulty of course assignments [12].

This article is organized as follows. In Section 2, we go over relevant background work on collecting participant data and then briefly present previous work on affect and socially inappropriate behavior. Next, in Section 3 we describe the context of the study, the data used for the study, and our research questions for this work. In Section 4 we first present our results and then discuss them in Section 5. Lastly, we conclude this work in Section 6.

2 BACKGROUND

Here, we first describe systems that are used to collect data for different analytics in the computing education research domain. Then, we discuss affect and socially inappropriate behavior.

2.1 Collecting participant data

This article focuses specifically on data from students' programming process, which has been stored for decades by systems that automatically assess students' solutions [11]. Initially, these systems stored submissions, but more recently, an increasing amount of systems that record the whole programming process have been developed [13]. These include systems that purposefully focus on

Koli Calling 2017, November 16–19, 2017, Koli, Finland

ACM ISBN 978-1-4503-5301-4/17/11…$15.00
https://doi.org/10.1145/3141880.3141902

learning to use the tools that developers use, online systems that are easy to access and require no downloading of software, and systems that are specifically designed for novices that highlight specific parts of a programming language.

Whilst open and semi-open data sets from students' working process exist [13], the majority of the data sets are closed to a degree. For example, the Blackbox-project [4] requires that the researchers who wish to use the data register and disclose research questions before gaining access.

From the legal perspective, each country and university has their own specific ethical rules that control how data can be collected and used. This is something that is likely to change especially within the European Union, as the new General Data Protection Regulation (GDPR) [7] becomes enforceable in May 2018. The GDPR takes a stance on the responsibilities of parties who either record, store or process data from individuals, and posit that permission for gathering data must be explicitly collected. Individuals must be able to decline data collection, and declining of data collection must have no detrimental consequences. Moreover, when asking for permission, explicit information on how the data will be used must be given, and individuals must be able to have their data removed.

2.2 Affect and socially inappropriate behavior

There exists only a handful of studies that have studied affect within the computing education domain. These include studies from Rodrigo et al. [19] who used observers who marked down instances of students' frustrations in a programming lab, and later studied whether programming process data could be used to detect frustration. Their results showed that a model built from consecutive syntax errors and the time that it takes to fix them had a weak correlation with frustration.

Similarly, fine-grained data has been used to determine stress [21], boredom and engagement [2], and emotional states in general [6]. In some studies, affective states have also been collected via retrospective reports. For example, Bosch et al. [3] studied novices' emotions and whether specific teaching interventions could alter them, whilst Ihantola et al. [12] used self-reports on assignment difficulty in conjunction with programming process data to automatically assess the difficulty of programming assignments – which can play a role in frustration, flow or boredom.

In our study, we specifically focus on cursing and swearing as socially inappropriate behavior. Cursing has been studied especially in the context of verbal fluency, where some studies have been motivated by the prejudice that cursing is often used to mask a lacking vocabulary. According to Jay and Jay, this is not true. They found that individuals who are fluent with taboo words such as swears are also overall more fluent verbally [15] – at the same time, whilst research indicates that individuals who swear fluently are verbally more fluent, one should not draw the conclusion that the use of profanities is linked with intelligence [8].

3 METHODOLOGY

3.1 Context

The data at our disposal comes from a massive open online course (MOOC) in programming. The course is given in English, expects no pre-requisite knowledge, and as is typical for MOOCs, anyone can attend the course without any type of a fee. The only expectation is that the participant should have a valid email address and a computer that he or she can use, and that the participant should be able to install a programming environment to this computer – a tutorial for the installation process is provided. Based on Google Analytics, the majority of the participants come from the United States of America, United Kingdom, Canada, India and Germany, but there are visits from over 100 separate countries.

The course material is organized as an online textbook with embedded assignment handouts. The content is divided into six logical parts where each part contains a set of materials and assignments. The MOOC is self-paced, and in order to proceed to the next logical part, the participant has to complete at least 85% of the assignments of the previous part. Each part has a dozen of programming assignments, all of which are automatically assessed.

During the course, the participants learn the principles of constructing object-oriented programs in Java, starting with input and output in terminal applications, and learning about loops, methods, lists, objects, and basic algorithms such as sorting and searching. All of the programming in the course is done within a custom programming environment, which can record data from the participants' programming process.

3.2 Data and preprocessing

As the participants work on course assignments, their programming process is recorded by the programming environment, given that the participants allow it. This means that subsequent program states are stored, and that the stored data can be used to reconstruct the programming process. Whilst the participants are encouraged to enable data gathering in the course for research purposes, it is not mandatory. The participants are free to enable or disable the recording at any time of the course, and they can also request for the deletion of all of their data.

For the present analysis, a subset of 10,000 participants who had completed at least three course assignments (out of approx. 100 assigments) was selected. For each participant, the final submissions for each assignment were extracted, and the working process on each assignment was reconstructed. To focus on deliberate writing, the working process data was filtered to include only sequential inserts. Finally, non-alphanumeric characters were replaced with blanks, and all uppercase words were transformed into lowercase for case insensitivity.

3.3 Research questions and analysis

The research questions in our study are as follows:

(1) How common is cursing while programming?
(2) Are there assignments in which cursing is more common?
(3) Do all participants curse alike?

For identification of cursing, a profane word list consisting of 86 profane words was manually constructed from a list of over 1300 potentially offensive English words[1]. These words were then identified from the participants' submissions and reconstructed working processes. To avoid false positives, words similar to what the students were expected to type – such as *hell* in "hello world" and *arse* in "parse" – were removed from the profane word list.

[1]The list of potentially offensive words is available at Luis von Ahn's research group page at CMU http://www.cs.cmu.edu/~biglou/resources/

Table 1: Percentage of participants who had curse words in submissions and snapshots, and the percentage of participants who were not observed to curse in a written format.

Cursing type	Percentage of participants (%)
Cursing in submissions	3.8%
Cursing in snapshots	14.3%
Not cursing	85.7%

To answer the first research question, "*How common is cursing while programming?*", exact matches of words in the profane word list were looked for in the participants' submissions and the reconstructed working process. To answer the second research question, "*Are there assignments in which cursing is more common?*", we analyzed each individual assignment and identified the proportion of participants who cursed in the specific assignment. Here, for each assignment, we only studied those participants who worked on the specific assignment.

Finally, to answer the third research question, "*Do all participants curse alike?*", we studied participants' tendencies to curse and analyzed if cursing is a stable phenomenon across the population, or if there exist a few individuals who are more prone to cursing than others. We also study whether cursing is linked to course completion by determining whether there is a link between the use of profanities and the quantity of completed assignments.

4 RESULTS

4.1 How common is cursing?

From the 10,000 participants in the data set, 3.8% used profanities in their submissions, and 14.3% used profanities as a part of their working process (Table 1). The snapshot data (data that participants do not explicitly submit for grading) shows that a significantly larger population writes profanities into their source code than one would be able to observe solely from submissions.

To seek a deeper understanding on why the participants curse, a manual analysis of a subsample of the participants' code was conducted. The analysis indicates that there is no single common approach in which the participants use profanities, but that profanities have multiple roles in the written programs. Profanities are used in variable names, comments and print statements – both in the role of debug statements as well as a part of the textual interfaces intended for the end user.

4.2 Is cursing assignment-specific?

To understand whether cursing could be explained to some degree by assignment-specific difficulty or other assignment-specific factors, we next analyzed the occurrence of cursing in individual assignments. For each assignment in the data, the solutions and working processes of those participants who worked on the assignment were analyzed, and the proportion of participants who cursed whilst working on the assignment was stored.

The proportion of students who used written profanities during the working process and in the submissions are plotted in Figure 1. For the majority of the assignments, less than 0.25% of the active students include profanities in the submitted assignments, whilst

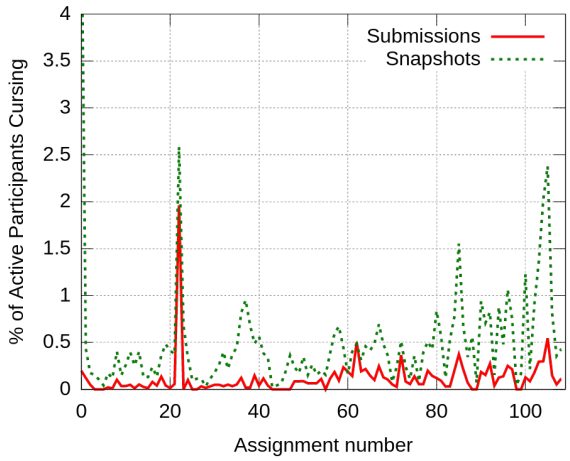

Figure 1: The proportion of participants who had worked on each assignment and written curse words in their submissions and snapshots. Submissions are marked using the red line, while snapshots are marked using the dotted line.

less than 1% of the active students use profanities while working on the assignments.

The data has a few noticeable spikes, however. The very first assignment, or "assignment 0", has a significant amount of profanities. The assignment is a sandbox assignment without any specified learning objectives or goals, and the students are not instructed to submit the assignment. That is, students are free to choose to not to do anything with the assignment – they are also free to try out whatever they wish with the assignment.

The second assignment that has a significant amount of profanities is the assignment 22, which is very similar to other assignments near to it. In the assignment, the students are expected to read input from the user, and compare the input to another variable. If the comparison succeeds, the system is supposed to print out a secret message. The course material, however, states that "*Your message can be whatever you want!*" – the data shows that such a note is an invitation of a bombardment of profanities.

The remaining of the obvious spikes are towards the end of the course. In these assignments, there is no obvious freedom in terms of students instructed to type any kinds of messages to the application. The assignments towards the end of the course are, however, more complex, and it is likely that they are more challenging for the students. It is possible that the observed increase in profanity in some of the latter assignments is due to increased difficulty. The analysis of difficulty, however, is out of the scope of this article.

4.3 Do all participants curse alike?

To understand how participants use profanities, we studied the vocabulary of each participant by counting each distinct profanity that the participant had written. The number of distinct profanities and the proportion of participants who used a range of distinct profanities are shown in Table 3. As can be observed from Table 3, majority of the participants who use profanities use only a few distinct profanities.

Table 2: The use of distinct profanities while programming categorized. The percentage of participants that fall into each category is reported.

Number of distinct profanities	% Participants
5 -	0.3%
3 - 4	1.1%
1 - 2	12.9%
0	85.7%

Table 3: The use of distinct profanities while programming categorized. The average number of assignments completed by participants in each category is reported.

Number of distinct profanities	Avg. completed assignments
5 -	85.5
3 - 4	71.2
1 - 2	71.8
0	47.3

Table 4: The mean and standard deviation of completed assignments for those who curse and those who do not.

	mean	st.dev.
Cursing	72.04	33.40
Non-cursing	47.34	37.12

The category with the most profanities – five or more distinct profanities – is the smallest, with only a handful of the participants. The largest vocabulary that an individual showed matched 16 of the profanities in our profanity word list.

When analyzing the cursing of participants with respect to their assignment completion rate, we observe that cursing seems to be linked to increased completion of assignments (Table 3). This means, that participants who do not curse at all complete less assignments, whilst the participants with the largest vocabulary also had the largest assignment completion rate. We decided to conduct a statistical analysis to study this in more detail. The students were split into two subpopulations: those who do not curse at all, and those who cursed at least once in the data. The mean and standard deviation of completed assignments for both groups is reported in Table 4.

We conducted a Kolmogorov-Smirnov test [16] to test whether the subpopulations are statistically significantly different. The results ($D = 0.31$, p-value < 0.0001) indicate that there is a statistically significant difference in regards to completed assignments for the two groups: those who curse tend to complete more assignments.

5 DISCUSSION

5.1 Profanities and performance

The data shows that the majority of the participants do not use any of the curse words that appear in our profanity list. Our analysis, however, only picks up exact matches of words, and for cursing to be detected, it has to be written. That is, the analysis does not detect individuals who curse verbally whilst working on course assignments.

There exists a small handful of participants who have a rather large cursing vocabulary – the largest such vocabulary consisted of sixteen distinct curse words. Most of the individuals who curse use only a few distinct curse words.

The individuals who curse at least once perform on average better than the individuals who do not curse at all. A statistical analysis of the correlation between the completed assignments and the quantity of curse words indicated that those who curse complete on average 50% more assignments on the course. However, it is possible that the difference is due to some people dropping out of the course before they cursed, but who would have cursed eventually at some point. For example, the assignment that asked students to type in any type of message had a lot more cursing when compared to other assignments. Thus, if a student dropped before that assignment, they were less likely to swear and had fewer completed assignments.

5.2 On the use of profanities

A visual analysis showed that profanities are used in multiple ways – they are used in comments, debug print statements, variable names, etc. The participants also use different types of profanities, ranging from mild profanities to racial slurs. Whilst the majority of the participants who cursed used profanities only marginally, some of the participants overflowed their work with profanities.

Profanities were more likely to appear in assignments without explicit learning objectives, assignments that explicitly stated that the participants can type in any kind of message that will be shown to the user, and – likely – in assignments that were more difficult than others. While in the first two cases, the use of profanities could be even categorized as joyful, in the third case profanities may have been used to vent frustration.

5.3 Reacting to profanities

Choosing to penalize each participant who used profanities in the course would mean penalizing between 4% and 14% of the participants, depending on whether private use of profanities is considered. If someone writes something offensive to the programming environment as they are programming, should we take action? Should we follow the way paved by Harvard [17]; kick the participants out and ban them from future courses? Or should we educate those individuals – for example, racist slurs may stem from ignorance and lack of education.

Not reacting to profanities can be taken as allowing the use of them, or even encouraging their use. In the case of racially offensive slurs, it could be meaningful to discourage their use through feedback. Whether it is something that the learning environment – or instructors – of a MOOC should do remains an open question.

5.4 On the collection and use of such data

While individuals use services and possibly agree to the use of their data, it is likely that individuals do not realize what is actually being collected. For example, many believe that Facebook stores

only full messages, while in reality texts that are not submitted are also collected [5].

It is also likely that the participants in this study – whilst agreeing to the use of their data for research purposes – do not realize what is being collected and that the data that they provide also includes profanities that they write. While we have studied swearing in source code, analogous analysis could be conducted to deduce e.g. political opinions from the way how individuals discuss in forums.

With the upcoming GDPR, entities that collect data are expected to explicitly state why data is collected. At the same time, research organizations have the possibility of having participants consent to specific scientific areas, "*...with recognised ethical standards for scientific research*" [7].

5.5 Limitations of the study

This study comes with several limitations, which we address here. First, as we only studied swearing in the English language and looked for words in a specific profane word list, it is likely that the actual proportion of participants who curse is larger. Some participants may have used their native language, some may prefer to curse using words not in our list, and some may simply have typos while cursing. Second, the study suffers from a selection bias: the course under study is a MOOC and we do not know if the results would apply also in a graded course offered at a University. Third, our course has not officially prohibited the use of profanities, which likely would influence the outcomes. On the other hand, prohibition might not be effective as has been found in e.g. psychology [22] and plagiarism [20].

6 CONCLUSIONS

In this article, we studied students' submissions in an online course and searched for the uses of profanity as an example of socially questionable behavior. While we have chosen to study programming process data in an approach that some may consider humorous, there is a real possibility that a similar analysis is conducted by someone else and that the results of the analysis are used to inform decisions pertaining to important aspects of an individual's life, e.g. university acceptance or hiring decisions.

A similar analysis as conducted here would be possible for detecting other information in addition to cursing such as uncovering political opinions, whilst we believe that the data source should be different from programming data.

Our results hint towards a trend that those who curse at least once while completing assignments also complete more assignments in total when compared to those who do not curse at all. However, future research should investigate whether this relationship is simply caused by those completing more assignments having more opportunities to curse and progressing further in the course where the more difficult assignments might incite more cursing.

Our future work consists of two separate streams. First, we are analyzing whether the use of profanities is an indicator of frustration or difficulty, and at the same time, we are looking to determine whether the use of profanities in programming is beneficial in terms of reducing pain and stress related to learning.

REFERENCES

[1] Alireza Ahadi, Raymond Lister, Heikki Haapala, and Arto Vihavainen. 2015. Exploring machine learning methods to automatically identify students in need of assistance. In *Proc. of the 11th Annual International Conference on International Computing Education Research*. ACM, 121–130.

[2] Robert Bixler and Sidney D'Mello. 2013. Detecting boredom and engagement during writing with keystroke analysis, task appraisals, and stable traits. In *Proc. of the 2013 international conference on Intelligent user interfaces*. ACM, 225–234.

[3] Nigel Bosch, Sidney D'Mello, and Caitlin Mills. 2013. What emotions do novices experience during their first computer programming learning session?. In *International Conference on Artificial Intelligence in Education*. Springer, 11–20.

[4] Neil Christopher Charles Brown, Michael Kölling, Davin McCall, and Ian Utting. 2014. Blackbox: A Large Scale Repository of Novice Programmers' Activity. In *Proc. of the 45th ACM Technical Symposium on Computer Science Education (SIGCSE '14)*. ACM, New York, NY, USA, 223–228. https://doi.org/10.1145/2538862.2538924

[5] Sauvik Das, Adam DI Kramer, et al. 2013. Self-Censorship on Facebook.. In *ICWSM*.

[6] Clayton Epp, Michael Lippold, and Regan L Mandryk. 2011. Identifying emotional states using keystroke dynamics. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 715–724.

[7] General Data Protection Regulation. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46. *Official Journal of the European Union (OJ)* 59 (2016), 1–88.

[8] Frank Giordano. 2016. The Relationship between Profanity and Intelligence. *Yale Review of Undergraduate Research in Psychology* (2016), 16.

[9] Glenn Greenwald and Ewen MacAskill. 2013. NSA Prism program taps in to user data of Apple, Google and others. *The Guardian* 7, 6 (2013), 1–43.

[10] Alex Hern. 2015. How to listen to (and delete) everything you've ever said to Google. https://www.theguardian.com/technology/2015/oct/13/google-voice-activity-listen-delete-recordings. (2015). Accessed: 2017-07-17.

[11] Petri Ihantola, Tuukka Ahoniemi, Ville Karavirta, and Otto Seppälä. 2010. Review of recent systems for automatic assessment of programming assignments. In *Proc. of the 10th Koli Calling International Conference on Computing Education Research*. ACM, 86–93.

[12] Petri Ihantola, Juha Sorva, and Arto Vihavainen. 2014. Automatically Detectable Indicators of Programming Assignment Difficulty. In *Proc. of the 15th Annual Conference on Information Technology Education (SIGITE '14)*. ACM, New York, NY, USA, 33–38. https://doi.org/10.1145/2656450.2656476

[13] Petri Ihantola, Arto Vihavainen, Alireza Ahadi, Matthew Butler, Jürgen Börstler, Stephen H Edwards, Essi Isohanni, Ari Korhonen, Andrew Petersen, Kelly Rivers, et al. 2015. Educational data mining and learning analytics in programming: Literature review and case studies. In *Proc. of the 2015 ITiCSE on Working Group Reports*. ACM, 41–63.

[14] Matthew C Jadud. 2006. Methods and tools for exploring novice compilation behaviour. In *Proc. of the 2nd international workshop on Computing education research*. ACM, 73–84.

[15] Kristin L Jay and Timothy B Jay. 2015. Taboo word fluency and knowledge of slurs and general pejoratives: Deconstructing the poverty-of-vocabulary myth. *Language Sciences* 52 (2015), 251–259.

[16] Hubert W Lilliefors. 1967. On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American statistical Association* 62, 318 (1967), 399–402.

[17] Hannah Natanson. 2017. Harvard Rescinds Acceptances for At Least Ten Students for Obscene Memes. http://www.thecrimson.com/article/2017/6/5/2021-offers-rescinded-memes/. (2017). Accessed: 2017-08-04.

[18] George Orwell. 1990. Nineteen Eighty-Four. 1949. *The Complete Novels* 7 (1990).

[19] Ma. Mercedes T. Rodrigo and Ryan S.J.d. Baker. 2009. Coarse-grained Detection of Student Frustration in an Introductory Programming Course. In *Proc. of the Fifth International Workshop on Computing Education Research Workshop (ICER '09)*. ACM, New York, NY, USA, 75–80. https://doi.org/10.1145/1584322.1584332

[20] Judy Sheard, Martin Dick, Selby Markham, Ian Macdonald, and Meaghan Walsh. 2002. Cheating and Plagiarism: Perceptions and Practices of First Year IT Students. In *Proc. of the 7th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE '02)*. ACM, New York, NY, USA, 183–187. https://doi.org/10.1145/544414.544468

[21] Lisa M Vizer, Lina Zhou, and Andrew Sears. 2009. Automated stress detection using keystroke and linguistic features: An exploratory study. *International Journal of Human-Computer Studies* 67, 10 (2009), 870–886.

[22] Daniel M Wegner and David J Schneider. 2003. The white bear story. *Psychological Inquiry* 14, 3-4 (2003), 326–329.

[23] Michael Zimmer. 2008. The gaze of the perfect search engine: Google as an infrastructure of dataveillance. *Web search* (2008), 77–99.